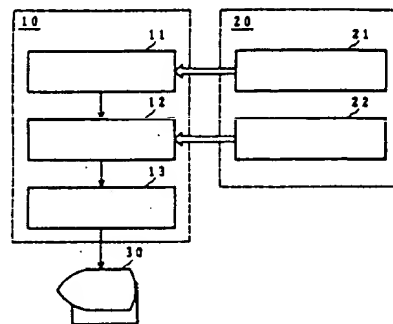


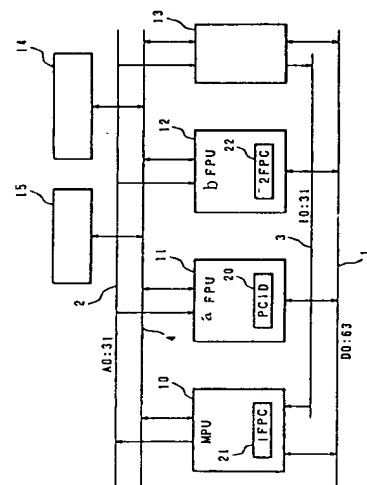
CONSTITUTION: Instruction information in the level of the output message with respect to the console describes in the job control language is stored in the first area 21 of a main storage device 20, and a console message text consisting of the output message with respect to the console and the level are stored in a second area 22. When the level of the output message included in the console message text inputted based on the message output instruction is more than instruction information of the level of the output message concerned, an output means 10 gives the output message to the console 30. Thus, the developing manhour and the number of the lines of the program can be reduced and the output waiting state of the console can be cancelled in a large-scale system, whereby the system can effectively be operated.



11: message level instruction information input means, 12: console message text input means, 13: console message text output decision means

(11) 2-306360 (A) (43) 19.12.1990 (19) JP
(22) Appl. No. 64-127148 (22) 19.5.1989
(71) MITSUBISHI ELECTRIC CORP (72) TOYOHICO YOSHIDA
(51) Int. Cl.⁵ G06F15/16, G06F9/38

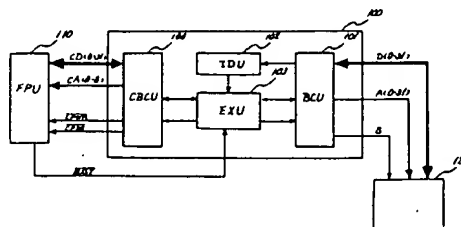
CONSTITUTION: When an operation is instructed to the coprocessor 11, the value of the program counter (PC) of the coprocessor instruction is not transferred, and the command connecting the entry number of a queue 21 storing the PC value and an operation indicator is transferred to the slave processor 11. The coprocessor 11 separates the entry number of the queue 21 and the operation indicator, and holds the entry number in a PCID queue 20. When an exception occurs at the time of executing the operation, exception information including the entry number of the corresponding PC queue 21 is continuously outputted from the coprocessor 11 and the PC value is read from the main processor 10. Thus, the memory cycle transferring the PC value is eliminated, and the instruction for the coprocessor instruction can be given at one memory cycle, whereby the data processing of high performance is attained.



13: memory system, 14: clock generator, 15: interruption control circuit, a: 1st, b: 2nd

(11) 2-306361 (A) (43) 19.12.1990 (19) JP
(21) Appl. No. 65-23507 (22) 31.1.1990 (33) JP (31) 89p.24858 (32) 3.2.1989
(71) NEC CORP (72) HIROAKI KANEKO
(51) Int. Cl.⁵ G06F15/16, G06F9/38

CONSTITUTION: A 32-bit microprocessor 100 is connected to a memory 120 through a bus control unit 101, the data bus D of 32-bit width, and the address bus A of 32-bit width. Besides, the coprocessor (FPU) 110 for floating point operation is connected to the CPU 100 through a slave processor bus control unit 104, the data bus CD of 32-bit width and the address bus CA of 9-bit width. Then, the data transfer with the coprocessor FPU 110 is executed by using the exclusive data bus CD and the exclusive address bus apart from the data bus D and the address bus A for accessing the memory 120. Thus, the duty cycle of the system bus at the time of the execution of an instruction to be expanded by the coprocessor can be reduced, and the occurrence of the bus neck can be suppressed.



⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

平2-306361

⑬ Int. Cl.⁹

G 06 F 15/16
9/38
15/16

識別記号

3 7 0 Z
3 7 0 C
4 0 0 K

庁内整理番号

6745-5B
7361-5B
6745-5B

⑭ 公開 平成2年(1990)12月19日

審査請求 未請求 請求項の数 1 (全9頁)

⑮ 発明の名称 マイクロプロセッサ

⑯ 特 願 平2-23507

⑰ 出 願 平2(1990)1月31日

優先権主張 ⑱ 平1(1989)2月3日 ⑲ 日本(JP) ⑳ 特願 平1-24858

㉑ 発 明 者 金子 博 昭 東京都港区芝5丁目33番1号 日本電気株式会社内

㉒ 出 願 人 日本電気株式会社 東京都港区芝5丁目7番1号

㉓ 代 理 人 弁理士 内 原 晋

明 細 書

1. 発明の名称

マイクロプロセッサ

2. 特許請求の範囲

命令セットを拡張するためのコプロセッサを外部に接続できるマイクロプロセッサにおいて、外部の間でデータ転送を行なうための端子群を2組備え、それぞれ端子群を独立に制御する専用の2組の制御手段を備え、メモリおよびI/Oに対するデータ転送を、前記第一の制御手段によって制御される第一の端子群により行ない、コプロセッサに対するデータ転送を、前記第二の制御手段によって制御される第二の端子群により行なうことを特徴とするマイクロプロセッサ。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、マイクロプロセッサが外部で処理で

きる命令の他に、マイクロプロセッサの外部に接続したコプロセッサによって命令を拡張することのできるマイクロプロセッサに関する。

〔従来の技術〕

マイクロプロセッサは、限られたサイズの半導体チップの上で実装する必要があるため、若干の性能を犠牲にしても独立した機能をチップ外部へ別なLSIとして実装することがしばしば行なわれる。それぞれの機能を単一の半導体チップに実装することで、物理的に製造不可能なチップ・サイズになったり、チップ・サイズの増加によるコスト増加を避けるためである。

特に、高性能、高機能を要求される16ビット以上のマイクロプロセッサでは、浮動小数点演算機能、メモリ管理機構、メモリ緩衝(キャッシュ)機構等の機能を有したLSIが、マイクロプロセッサと密に接続する形で用いられる。これらの機能は、本来はマイクロプロセッサの内部に実装することで、より高性能、システムの小型化が可能になるものであるが、前述の理由により、

マイクロプロセッサ外部で実現されている。

マイクロプロセッサ外部にあって、マイクロプロセッサの命令セットを拡張するための機能を持つLSIは、一般にコプロセッサとして知られている。コプロセッサの代表的な機能として、前にも述べた浮動小数点演算がある。

従来の4, 8, および幾つかの16ビット・マイクロプロセッサは、応用分野が単純であったこともあり、整数を扱う命令しか有していなかった。しかしながら、マイクロプロセッサの応用分野の拡大に伴い、浮動小数点演算を高速に実現することが要求されつつある。従来のマイクロプロセッサでは、浮動小数点演算用の命令を持たないために、整数を取り扱う命令を組合せ、ソフトウェアによって浮動小数点演算を行っていた。このため、整数演算と比較して浮動小数点の演算は、極めて低い性能しか持ち合わせていなかった。

浮動小数点演算用コプロセッサは、マイクロプロセッサと密に接続され、浮動小数点演算用の命令をマイクロプロセッサが実行しようとする（実

際には演算できない）ことを、マイクロプロセッサから通知、あるいは自ら検知すると、マイクロプロセッサにかわり浮動小数点演算を行なうものである。浮動小数点演算用コプロセッサが浮動小数点演算を行なっている間、マイクロプロセッサは浮動小数点演算命令以外の命令を続行することができる。

第3図に従来のマイクロプロセッサと、浮動小数点演算用コプロセッサの接続例を示す。32ビット・マイクロプロセッサ（以下CPUと呼ぶ）300は、32ビット幅のデータ・バスD、32ビット幅のアドレス・バスAを介して、浮動小数点演算用コプロセッサ（以下FPUと呼ぶ）310、ならびにメモリ320と接続されている。ステータス信号群Sは、CPU300が発行するバス・サイクルの種類、およびタイミングをFPU310、ならびにメモリ320に通知するための信号群である。BUSY信号は、FPU310が浮動小数点演算を実行中であることをCPU300に通知するための信号である。

CPU300は、バス・サイクルの発生を制御するバス制御ユニット（以下BCUと呼ぶ）301、BCU301によって取り込まれた命令コードを解説する命令デコード・ユニット（以下IDUと呼ぶ）302、および実際に命令をする実行ユニット（以下EXUと呼ぶ）303によって構成される。

まず、浮動小数点演算命令を除く命令（以下通常命令と呼ぶ）を実行する場合のCPU300の動作について第4図とともに説明する。命令フェッチ・バス・（FETCH）サイクルによってメモリ320からBCU301に取り込まれた命令コードは、IDU302によって解説され、動作の種類、オペランドの種類（ex. レジスタ／メモリ、リード／ライト等）がEXU303に通知される。

EXU303は、メモリ・リード・オペランドがある場合BCU301にオペランド・リード・バス・（MEMR）サイクルの起動を指示する。メモリ320に格納されているメモリ・オペラン

Dは、MEMRサイクルによってBCU301に転送される（読み出される）。レジスタ・オペランドがある場合は、図面では省略しているが、EXU303内の汎用レジスタがオペランドとして選択される。

メモリ・オペランドがBCU301に用意されると、このデータを引き取ってEXU303は実際の命令実行を行なう。

EXU303は、命令実行結果がメモリ320に書き戻す必要があることをIDU302によって通知されていれば、BCU301にオペランド・ライト・バス（MEMW）サイクルの起動を指示すると同時に、BCU301に対して書き戻すべきデータ（実行結果）を転送する。メモリ320には、MEMWサイクルによって、必要な実行結果が転送される（書き込まれる）。

一方、浮動小数点演算命令を実行する場合、IDU302によって動作の種類、オペランドの種類がEXU303に通知されるまでは同一である。

メモリ320に格納されている浮動小数点デー

タは、一般に32ビット以上(64ビットあるいは80ビット)であるため、1回のMEMRサイクルではBCU301に取り込むことはできず、複数のバス・サイクルが必要である。EXU303は、BCU301浮動小数点データをすべて取り込むと、これをFPU310に対して転送するために、コプロセッサ・オペランド・ライト・バス(CPOW)サイクルの起動を指示し、BCU301内に一時格納された浮動小数点データは、FPU310に転送される(書き込まれる)。この転送も、前記理由により、複数回のバス・サイクルが必要なことが多い。

IDU302によってEXU303に通知された命令実行に関する動作の種類は、EXU303によってFPU310が解説できるような形式(コマンド)に再構成される。EXU303で生成されたコマンドは、BCU301に転送されると同時に、EXU303はBCU301に対してコプロセッサ・ライト・バス(CPWR)サイクルの起動を指示する。

む)ように、BCU301に対して指示を行なう。
〔発明が解決しようとする課題〕

前述した浮動小数点命令のCPU300が実行する際に、CPU300が発行するバス・サイクルによるデータ・バスDの状態を第4図に示す。ここでは、簡便のために浮動小数点データのビット幅は32ビット、すなわちCPU300とメモリ320、およびCPU300とFPU310の間で行なわれる浮動小数点データの転送は、1回のバス・サイクルで十分であるとする。

浮動小数点演算命令の実行をCPU300からFPU310へ分離したことにより、本来CPU300にとって不要なCPOW、CPWR、およびCPORの各サイクルが発行されることが簡単に理解できる。先に述べたように、CPOW、CPORの各サイクルは、一般に2回以上であることが多いため、本来無駄なバス・サイクルは実質的にはさらに多い。

また、2つのオペランド・データに対する演算(例えば加算)を行なう場合に両オペランドが共

CPWRサイクルによってコマンドがFPU310に転送される(書き込まれる)と、FPU310はコマンドに先立って転送された浮動小数点データに対して演算を開始する。

FPU310が演算中は、BUSY信号によってビジー状態であることがCPU300に対して通知され、EXU303は浮動小数点演算命令の実行が終了しておらず、引続く命令の実行を開始できないことを検知して、BUSY信号がレディー状態になることを待ち合わせる。

FPU310が演算を終了すると、BUSY信号はレディー状態になり、EXU303によって演算終了が検知される。通常命令と同様に、演算結果がメモリ320に書き戻す必要があることをIDU302によって通知されていれば、EXU303はコプロセッサ・オペランド・リード・バス(CPOR)サイクルを起動し、一度FPU310内の演算結果をBCU301内に転送(読み出す)した後、メモリ320にMEMWサイクルによって、必要な実行結果を転送する(書き込

にメモリ・オペランドの際には、第一のオペランド、および第二のオペランドに対して、それぞれCPOWサイクルを必要とすることになる。

例えば64ビット長の2つのメモリ320に格納された浮動小数点データに対して、浮動小数点演算を行なう場合、4回のCPOWサイクル、1回のCPWRサイクル、2回のCPORサイクル、合計7回のバス・サイクルがFPU310を用いるために必要になる。

近年マイクロプロセッサは、高性能化を図るために、命令のフェッチ、命令のデコード、命令の実行、あるいはさらに段数を増やした、同時に複数の命令を処理するパイプライン方式を採用している。

パイプライン方式の利点は、命令処理をいくつかのステージに分割することで、見かえ上の実行性能を向上できることにある。しかしながら、この利点は分割したステージで観みなく命令処理が行なわれ、初めて得られるものである。

前段階のステージで、後段階のステージの結果

を必要とするような、いわゆるパイプライン・ハザードの発生時や、パイプラインの根本である外部データ・バスが使用中に必要なバス・サイクルを発行できないようなバス・ネック状態では、理想的な性能を得ることはできない。

前述したように、コプロセッサを用いるために、マイクロプロセッサ、メモリ、そしてコプロセッサが共有するアドレス・バス、データ・バスを一定時間占有するために、バス・ネック状態の発生を助長することになる。

特に、コプロセッサ側の処理能力が優れている場合、例えば従来の浮動小数点用コプロセッサでさえ、頻繁に使用される四則演算を10クロック程度で処理できるにもかかわらず、データ・バス使用のためのクロック数が上回る状態にさえ成り得る。

このような状態では、マイクロプロセッサが被るオーバーヘッドは、深刻である。

〔課題を解決するための手段〕

本発明は、命令セットを拡張するためのコプロ

セッサを外部に接続できるマイクロプロセッサにおいて、外部とデータ転送を行なうためのデータ・バス、アドレス・バス、および状態信号群をドライブする端子を2組備え、それぞれの端子群を独立に制御する2組の専用の制御手段を有することを特徴とする。

すなわち、本発明ではメモリをアクセスするためのデータ・バス、およびアドレス・バスとは別に、専用のデータ・バス、およびアドレス・バスを用いてコプロセッサとデータ転送を行なっている。

〔実施例〕

以下図面を参照して、本発明の構成および動作を詳細に説明する。

第1図は、本発明を用いたマイクロプロセッサと、浮動小数点演算用コプロセッサの接続例を示す。32ビット・マイクロプロセッサ（以下CPUと呼ぶ）100は、32ビット幅のデータ・バス（0-31）、32ビット幅のアドレス・バスA（0-31）を介してメモリ120と接続されている。ステータス信号群Sは、CPU100が

発行するバス・サイクルの種類、およびタイミングをメモリ120に通知するための信号群である。データ・バスD、アドレス・バスA、およびステータス信号群Sを総称してシステム・バスと呼ぶ。

システム・バスで発行されるバス・サイクルは、リード／ライトともに2クロックで構成される。図面では省略しているが、メモリ120から要求されるバス・サイクルの延長要求信号READYによって、2クロック以上に延長することができる。

また浮動小数点演算用コプロセッサ（以下FPUと呼ぶ）110は、32ビット幅のデータ・バスCD（0-31）、9ビット幅のアドレス・バスCA（0-8）を介してCPU100と接続されている。

ステータス信号群としてライト信号WR（-）およびストローブ信号DS（-）がCPU100からFPU110に出力される。ステータス信号群WR（-）、DS（-）は、CPU100が発行するバス・サイクルの種類、およびタイミングをFPU

110に通知するための信号群である。データ・バスCD、アドレス・バスCA、およびステータス信号群CPWR（-）、CPDS（-）を総称してコプロセッサ・バスと呼ぶ。BUSY（-）信号は、FPU110が浮動小数点演算を実行中であることをCPU100に通知するための信号である。

CPU100は、システム・バスのバス・サイクル発生を制御するバス制御ユニット（以下BCUと呼ぶ）101、BCU101によって取り込まれた命令コードを解読する命令デコード・ユニット（以下IDUと呼ぶ）102、および実際に命令をする実行ユニット（以下EXUと呼ぶ）103、コプロセッサ・バスのバス・サイクル発生を制御するバス制御ユニット（以下CBCUと呼ぶ）104によって構成される。

CBCU104は、EXU103の指示により、コプロセッサ・バスに対してバス・サイクルを発生し、データ・バスCDを介してFPU110とデータの転送（読出し／書き込み）を行なうことができる。FPU110へ転送する（書き込む）

データは、EXU103からCBCU104へ設定され、FPU110から転送する（読み出す）データは、CBCU104からEXU103へ取り込まれる。

CBCU104で制御され、コプロセッサ・バスで発行されるバス・サイクルは、第6図に示すようにリード／ライトともに3クロック（C1、C2、C3の各ステート）で構成される。

バス・サイクルを延長する制御は行われず、必ず3クロックで終了する点がシステム・バスにおけるバス・サイクル制御と大きく異なる。バス・サイクルの延長制御を持たないため、CBCU104はBCU101よりも簡単に実現できるとともに、CPU100とFPU110の間のインタフェースを回路的、およびタイミング的に容易にしている。

FPU100からCPU110へのリードを行う場合、アドレス・バスCA（0-8）で指定されるFPU100内のレジスタの内容は、データ・バスCD（0-31）を介してC3ステートの

終了時にCBCU104に取り込まれる。CPU100からリード・バス・サイクルが起動されたことを、FPU110はCPWR(-)信号がHighであることを、CPDS(-)信号がLowに立下ったことで検知する。一方、FPU100へCPUからのライトを行う場合、CBCU104からデータ・バスCD（0-31）に出力されるデータは、C3ステートの終了時にアドレス・バスA（0-8）で指定されるFPU100内のレジスタに書き込まれる。CPU100からライト・バス・サイクルが起動されたことを、FPU110はCPWR(-)信号がLowであることを、CPDS(-)信号がHighに立上ったことで検知する。

アドレス・バスCA（0-8）によって、最大512ケのFPU110内の異なったレジスタに対してリード／ライトを行うことができる。

次に、本実施例の具体的な動作について説明する。

まず、通常命令を実行する場合は、従来のマイクロプロセッサと全く同様に動作するので、説明を省略する。

浮動小数点演算命令を実行する場合は、説明を簡便にするために、浮動小数点データは32ビット長であり、対象となるオペランドはメモリ120にあるものとする。例としては、加算を行うFADD、Sreg、mem命令がある。この命令は、メモリ120に格納された32ビット浮動小数点データとFPU110内のデータ・レジスタの値を加算して、メモリ120に結果を書き戻すものである。

IDU102によって動作の種類、オペランドの種類がEXU103に通知されるまでは、すなわちオペランド・アドレスの計算や命令デコード処理は従来と同一である。

メモリ120に格納されている浮動小数点データは、EXU103の指示により発行されたMEMRサイクルによってBCU101に転送（読み込み）されると、EXU103は該データをBCU101からCBCU104へ転送するとともに、CBCU104に対してCPOWサイクルの起動を指示する。

このCPOWサイクルについて、アドレス・バスCAにはFPU110内のデータ・ポートを選択するための値を出力しておく。

CPOWサイクルによって、浮動小数点データはFPU110の転送（書き込み）される。

従来どおり、IDU102によってEXU103に通知された命令実行に関する動作の種類は、EXU103によってコマンドに再構成される。EXU103で生成されたコマンドは、CBCU104に転送されると同時に、EXU103はCBCU104に対してCPWRサイクルの起動を指示する。

このCPUOWサイクルについて、アドレス・バスCAにはFPU110内のコマンド・ポートを選択するための値を出力しておく。

CPWRサイクルによってコマンドがFPU110に転送される（書き込まれる）と、FPU110はコマンドに先立って、データ・ポートに転送された浮動小数点データ、コマンド・ポートに転送されたコマンドに含まれる指定によって選

扱されるデータ・レジスタに対して、浮動小数点演算を開始する。

BUSY(-)信号の待合せは、従来と同様にEXU103によって行なわれる。

FPU110における演算が終了すると、演算結果はFPU110のデータ・ポートに設定されると同時に、BUSY(-)信号がHighに設定される。

FPU110が演算を終了すると、演算結果がメモリ120に書き戻す必要があることをIDU102によって通知されているので、EXU103はCPORサイクルを起動し、一度FPU110内の演算結果をCBCU104内に転送する（読み出す）。このCPORサイクルについて、アドレス・バスCAにはFPU110内のデータ・ポートを選択するための値を出力しておく。

EXU103は該実行結果をBCU101に転送すると同時に、メモリ120にMEMWサイクルによって、必要な実行結果を転送する（書き込む）ように、BCU101に対して指示を行なう。

712は、CNT703の発生する制御信号により、EBUS上の下位9ビットのデータをラッチすることができる。

CPBCNT712は、CNT703の発生する制御信号によりコプロセッサ・バス・サイクルのシーケンスを発生し、DS(-)、WR(-)信号を発生するとともに、バス・サイクルの種類によってCPDR711、CPAR712の端子制御、ならびにCD(0-31)からCPDR711へのラッチ・タイミングを発生する。

BUSY(-)信号は、他の状態信号と同様にCNT703に入力され、CNT703のシーケンス制御に関与する。

なお、FPU110に転送した浮動小数点データ、コマンドがあらかじめ定められたフォーマットに違反した場合、あるいはFPU110における演算中にオーバーフローなど例外が発生した場合、図面では省略したがFPU110から例外を検出した信号ERRによってEXU103に例外が通知される。EXU103はERR信号によって例

第7図に本発明の構成の中心となるEXU103、およびCBCU104の一構成例を示す。

EXU103およびCBCU104は、データ転送を行うための32ビット内部データ・バスEBUSで結合される。EXU103には、汎用レジスタ（以下GRと呼ぶ）701、二進ALUを含む演算器（以下ALUと呼ぶ）702、EXU103内および外部の資源に制御信号を発生し、状態信号によりシーケンス制御を行う制御部（以下CNTと呼ぶ）703で構成される。CNT703はマイクロプログラムによって実現されることが一般的である。

CBCU104は、32ビットの双方向性ラッチ（以下CPDRと呼ぶ）711、9ビット・ラッチ（以下CPARと呼ぶ）712、コプロセッサ・バス・サイクル制御部CPBCNT713で構成される。CPDR711は、CNT703の発生する制御信号により、EBUS上のデータをラッチするとともに、CPDR711の内容自体をEBUSに転送することができる。CPAR

外を検知した場合、例外の種類を知るために、FPU110内にありステータスを示すステータス・ポートをデータ・バスCDを経由して転送して（読み込み）、例外処理を行う。

前述した浮動小数点命令をCPU100が実行する際に、CPU100が発行するシステム・バス、およびコプロセッサ・バスにおけるバス・サイクルによるそれぞれのデータバスD、およびCDの状態を第5図に示す。

FPU110に対する浮動小数点データ、コマンド、および演算結果の転送のためのCPOW、CPWR、CPORサイクルが、コプロセッサ・バスで発行されるため、従来のデータ・バス状態を示す第四図と比較して、システム・バスに空き状態が増えていることが理解できる。

CPU100は、浮動小数点演算命令に引続く命令の命令コード・フェッチや、メモリ・オペランドのアクセスにこの期間を利用することができる。

本実施例では、レジスタ・メモリ間のオペランド演算を行う場合について説明した。

この場合、コプロセッサ・バスで3回のデータ転送が行われる。前述のようにシステム・バスを用いた場合、少なくとも6クロック(3回×2クロック)のロスを生じることになる。代表的なFPUでは、32ビットの浮動小数点演算は8クロック程度で処理することができるため、転送に要する6クロックの影響は本来の性能を60%に低下させてしまうことになる。これを100%発揮できる本発明の効果は明らかである。

さらに、メモリー・メモリー間で行う64ビットの浮動小数点演算では、3回×2バス・サイクル×2クロック(合計12クロック)の転送時間が必要であり、本発明を用いない場合、40%の性能しか得られないことになる。

なお、実施例で述べたようにコプロセッサ・バスの制御にはバス・サイクルの延長を行う必要がないため、コプロセッサへのアクセスを検知してバス・サイクルを延長させる外部回路が不要になる。プロセッサの動作速度が速まり、コプロセッサ・バスのバス・サイクルがコプロセッサへのア

クセスには高速すぎる場合には、固定クロック数バス・サイクルが延長することを選択する入力端子をCPU100に設け、CBCUによってバス・サイクルを延長すれば良い。

次に、本発明の別の実施例について説明する。

上記実施例では、コプロセッサ・バスに接続されるFPUは単一であったが、本実施例ではチップ・セレクト端子CSを有したFPU201、202が接続されていることが特徴的である。

FPU201、202のCS端子は、コプロセッサ・バスに発行されるバス・サイクルがそれぞれのFPUに対して有効であることを指定するために使用される。

それぞれのFPUのCS端子へは、アドレス・バスCAの一部の内容をデコードし、FPU201を選択する信号CS0(-)、およびFPU202を選択する信号CS1(-)を発生するデコーダ203によりチップ・セレクト信号が供給される。デコーダ203は、アドレス・バスCAの値により、CS0(-)信号、CS1(-)信号を独立に発生

することも、また同時に発生することもできる。

FPU201、FPU202のBUSY(-)出力信号は、ワイアード接続され、CPU200に取り込まれ、FPU201、FPU202のいずれかがビジー状態であれば、CPU200にビジー状態が通知される。

抵抗R204は、共有するBUSY(-)信号のHighレベルを保证するためのプルアップ抵抗である。

本実施例では、1つのコプロセッサ・バス上に複数のFPUを用意することで、同時に異なった浮動小数点演算を行ったり、異なった浮動小数点データに対して同一の浮動小数点演算を行なう並行動作により、前の実施例の構成より浮動小数点演算能力を向上させることができる。

〔発明の効果〕

以上説明したように、本発明を用い、コプロセッサへのデータ転送をシステム・バスから分離することで、コプロセッサで拡張される命令実行時のシステム・バスの使用率を低減させ、バス・

ネックの発生を抑えることができるため、パイプラインの各ステージでの命令処理をスムーズに行なうことにより、よりオーバーヘッドが少ない命令実行が可能な情報処理システムが実現できる。

また、前述したようにコプロセッサが専用使用するバス(コプロセッサ・バス)は、システム・バスと相似な構成をしているために、従来はシステム・バスに接続していたコプロセッサを変更することなしに、またインタフェース回路もそのまま使用できるという利点がある。

さらに、コプロセッサ・バスは、メモリーや入出力装置等の大量のバス資源を接続しなければならないシステム・バスとは異なり、コプロセッサのみ接続するため、ドライブ電流や、負荷容量の点で規模が小さくなるため、コプロセッサにとっては電気的結合上のインタフェースが容易になるという効果も得られる。

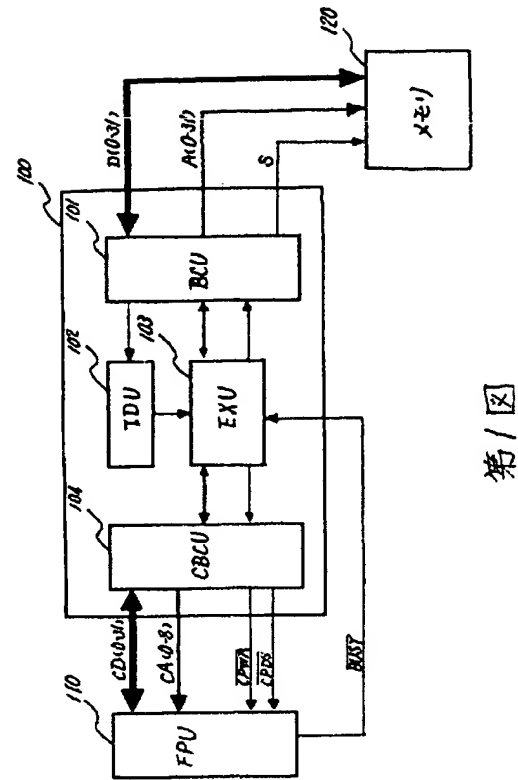
4. 図面の簡単な説明

第1図は本発明の一実施例の構成を示すブロッ

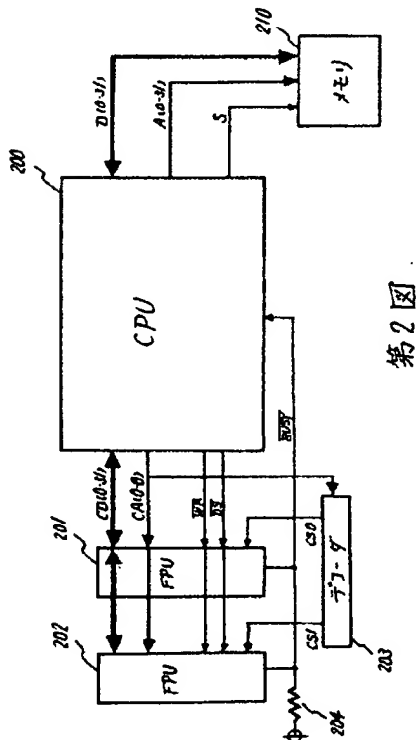
ク図、第2図は本発明の他の実施例の構成を示すブロック図、第3図は従来のCPUとFPUの接続構成例を示すブロック図、第4図は従来例におけるバス・サイクルの状態図、第5図は第1図におけるバス・サイクルの状態を示す図、第6図はコプロセッサバス・サイクルタイミング図、第7図は第1図のEXUおよびCBCUの一構成ブロック図である。

100、200、300……マイクロプロセッサ、110、201、202、310……浮動小数点演算用コプロセッサ、120、219、320……メモリ、101、301……バス制御ユニット、102、302……命令デコード・ユニット、103、303……命令実行ユニット、104……コプロセッサ・バス制御ユニット、203……デコーダ、204……プルアップ抵抗。

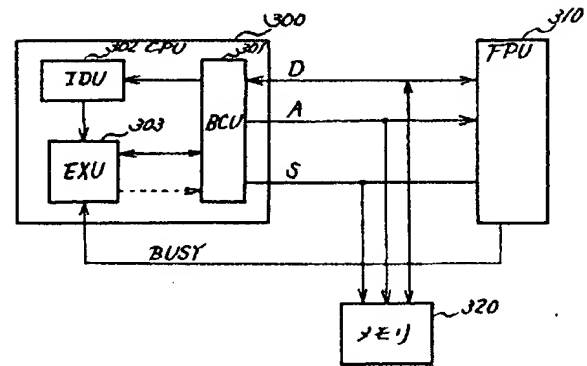
代理人 弁理士 内 原 晋



第1図



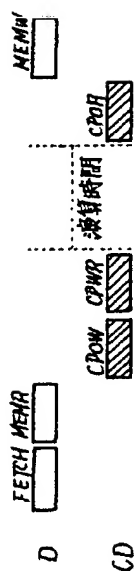
第2図



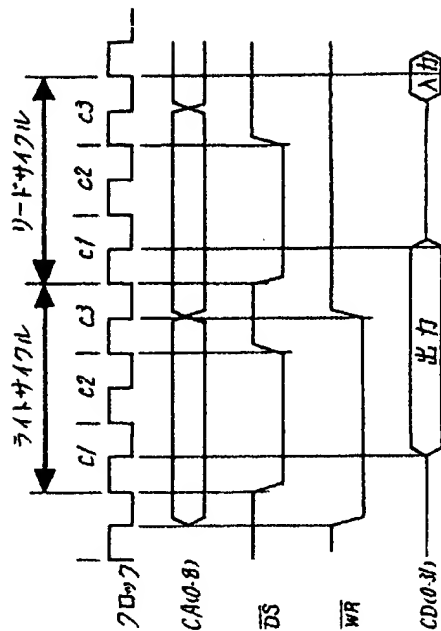
第3図



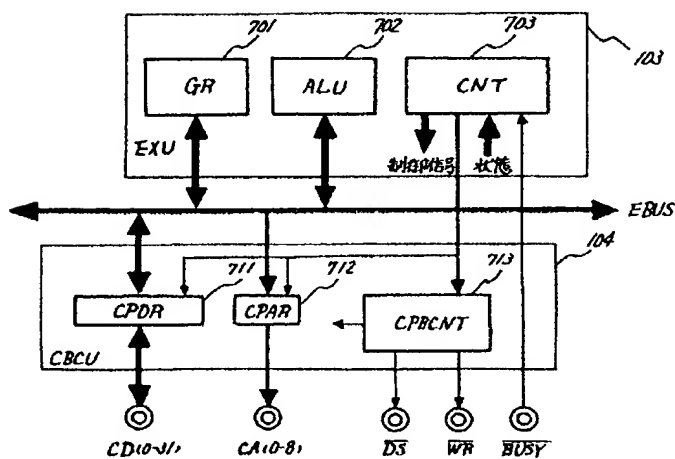
第4図



第5図



第6図



第7図

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.